

# The Why and How of Wiki Farms

---

**Dr. Cindy Cicalese**

**SMWCon Spring 2015**

# Agenda

---

- **What is a wiki farm?**
- **Benefits of wiki farms**
- **Goals of Gestalt wiki farm architecture**
- **MITRE's use of Gestalt wiki farms**
- **How we build a Gestalt wiki farm**
- **Future directions**

---

# Why?

---



# What is a wiki farm?

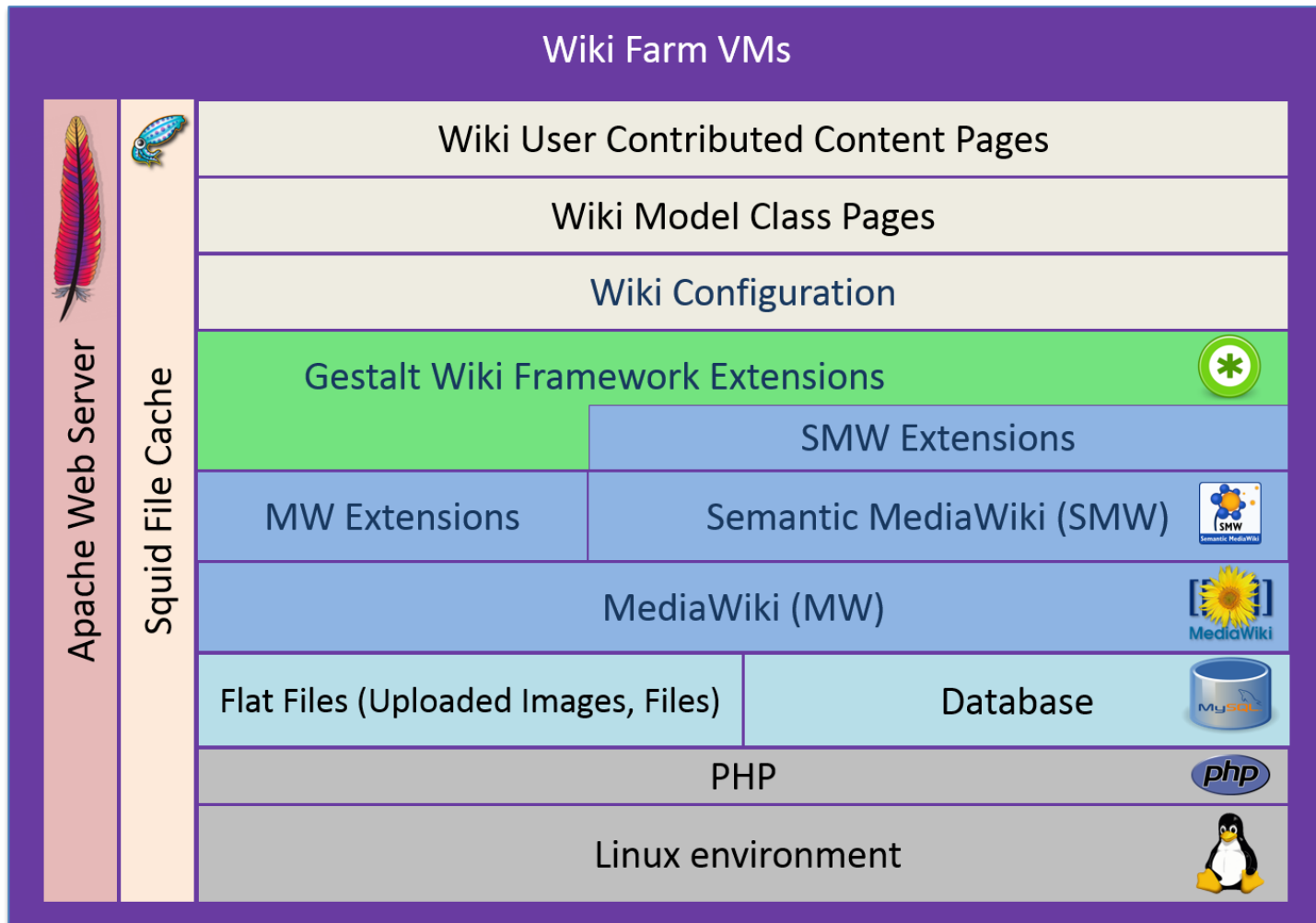
- **My definition:**
  - A MediaWiki server installation where the core software and (some) extensions are shared by a group of otherwise independent wikis
- **“wiki farm” or “wiki family”?**
- **My definition does not necessarily imply:**
  - commercial hosting
  - lack of command-line access
- **There is not a single way to create a MediaWiki wiki farm/wiki family**
  - Some alternatives: [http://www.mediawiki.org/wiki/Wiki\\_farm](http://www.mediawiki.org/wiki/Wiki_farm)  
(which redirects to [http://www.mediawiki.org/wiki/Manual:Wiki\\_family](http://www.mediawiki.org/wiki/Manual:Wiki_family))

# Benefits of wiki farms

- **Allows you to provision a new wiki very quickly**
- **Simpler configuration management**
  - Upgrade all wikis at once
  - But: cannot run different software versions for different wikis easily
- **Make good use of hardware/virtual resources**
- **Make good use of systems administrators' time**
- **Can be easy to replicate for testing/development purposes**



# Gestalt Wiki Framework



**The Gestalt Wiki Framework (GWF) is a set of patterns and open source software that enables groups to build enterprise knowledge using structured wikis**

# Gestalt wiki farm design decisions

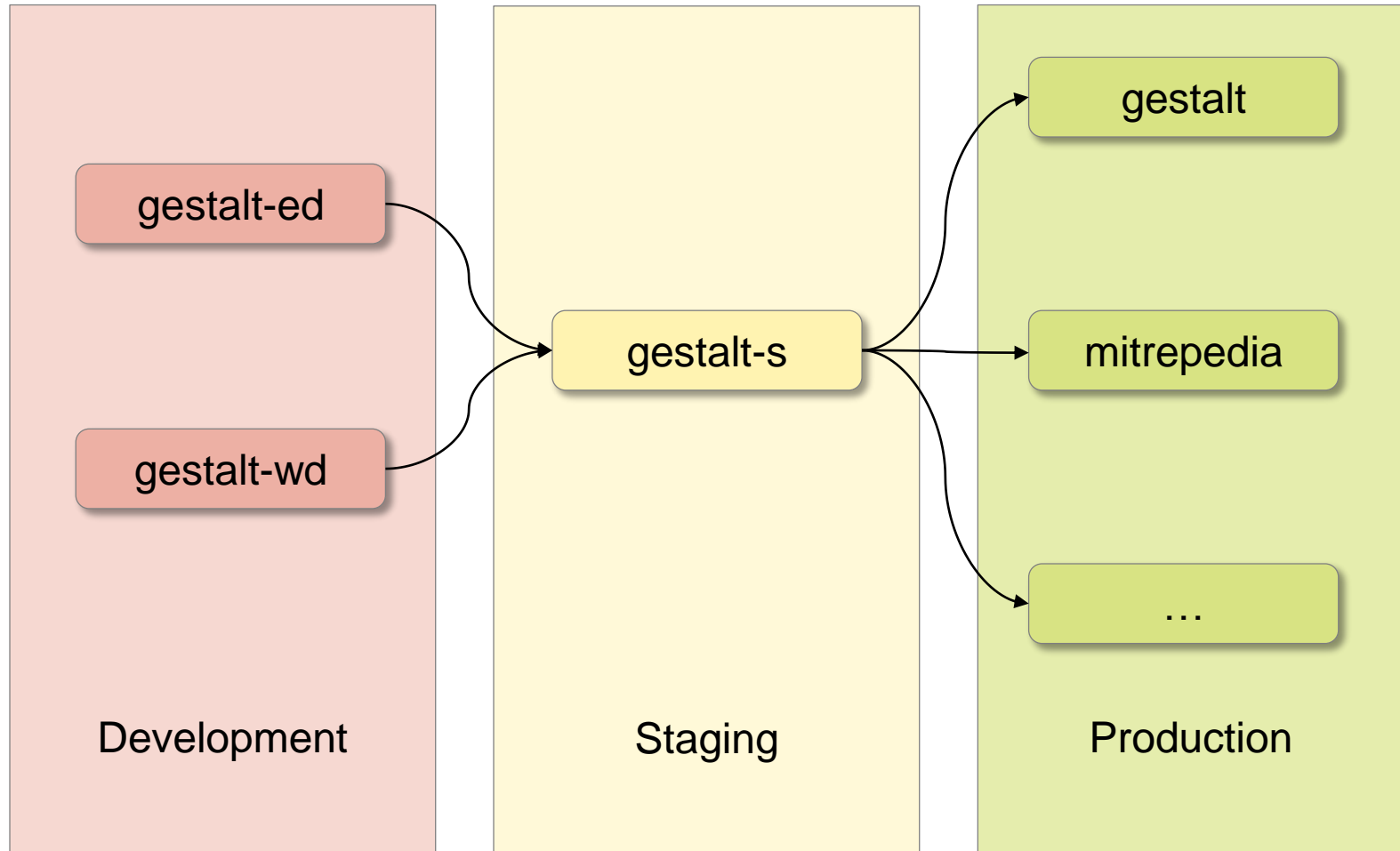
- **Linux/Apache/MySQL/PHP (usually)**
  - Usually RedHat or CentOS
- **Squid for caching/acceleration**
- **Everything that does not need to be in /etc lives in a single directory hierarchy**
  - Can be tarred and cloned to another host easily
  - Includes MySQL data files (assumes MySQL only for use of wikis)
- **Wiki customization**
  - Custom extensions
  - Custom namespaces
  - Custom configuration variables

# MITRE's use of Gestalt wiki farms

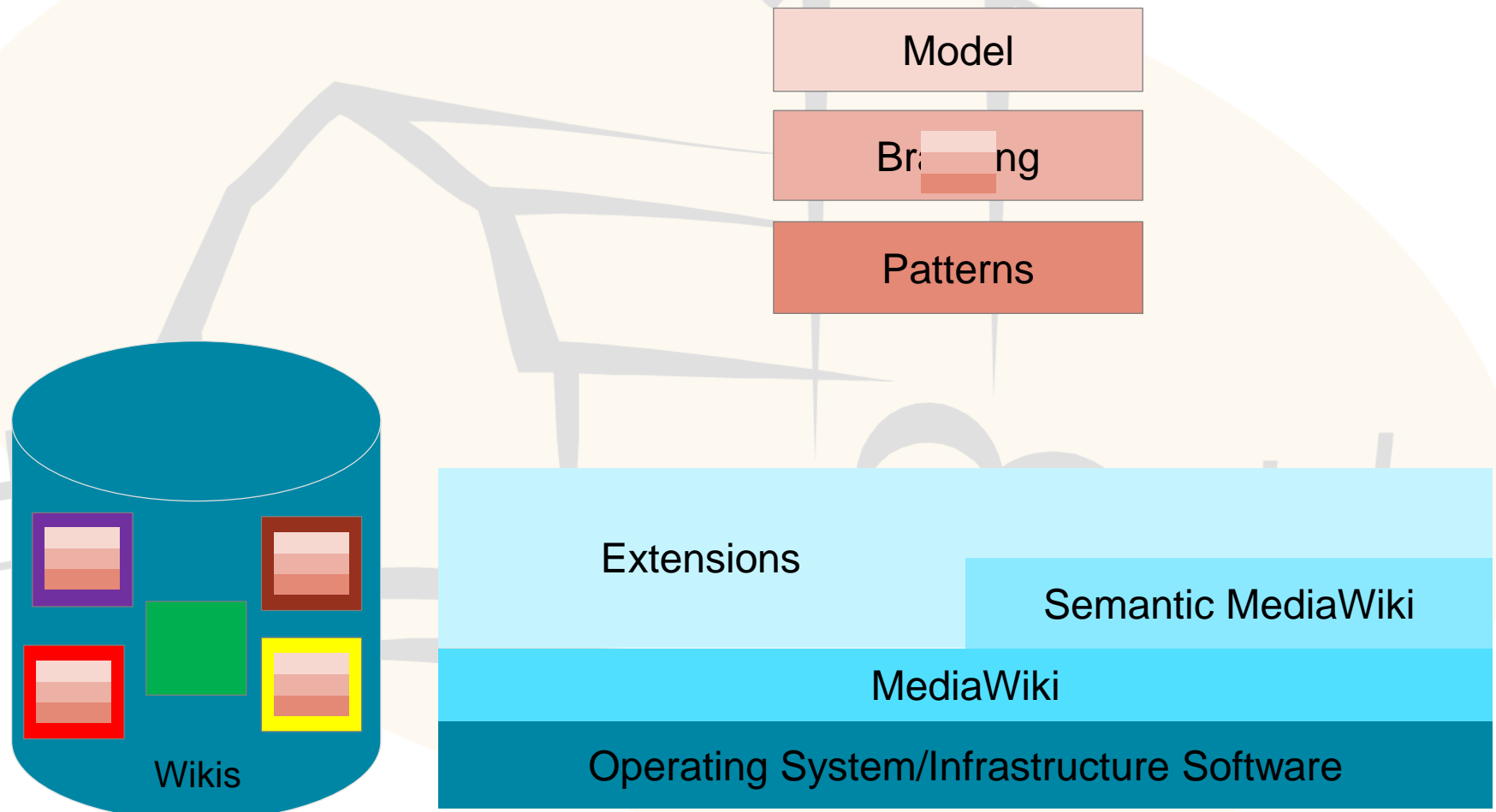
- **Inside the firewall:**
  - Several production wiki farms
  - Two staging wiki farms (internal and external)
  - Two development wiki farms (wiki development and extension development)
- **Outside the firewall – MITRE Partnership Network (MPN):**
  - Must have MPN account and be authorized to log in to wiki
  - Partnership wiki farm
    - Must log in to read and to edit
  - Open wiki farm
    - Open for reading
    - Must log in to edit
- **Customer wiki farms**



# Gestalt wiki development lifecycle



# Gestalt wiki farm deployment and maintenance



---

# How?

---

# How we build a Gestalt wiki farm

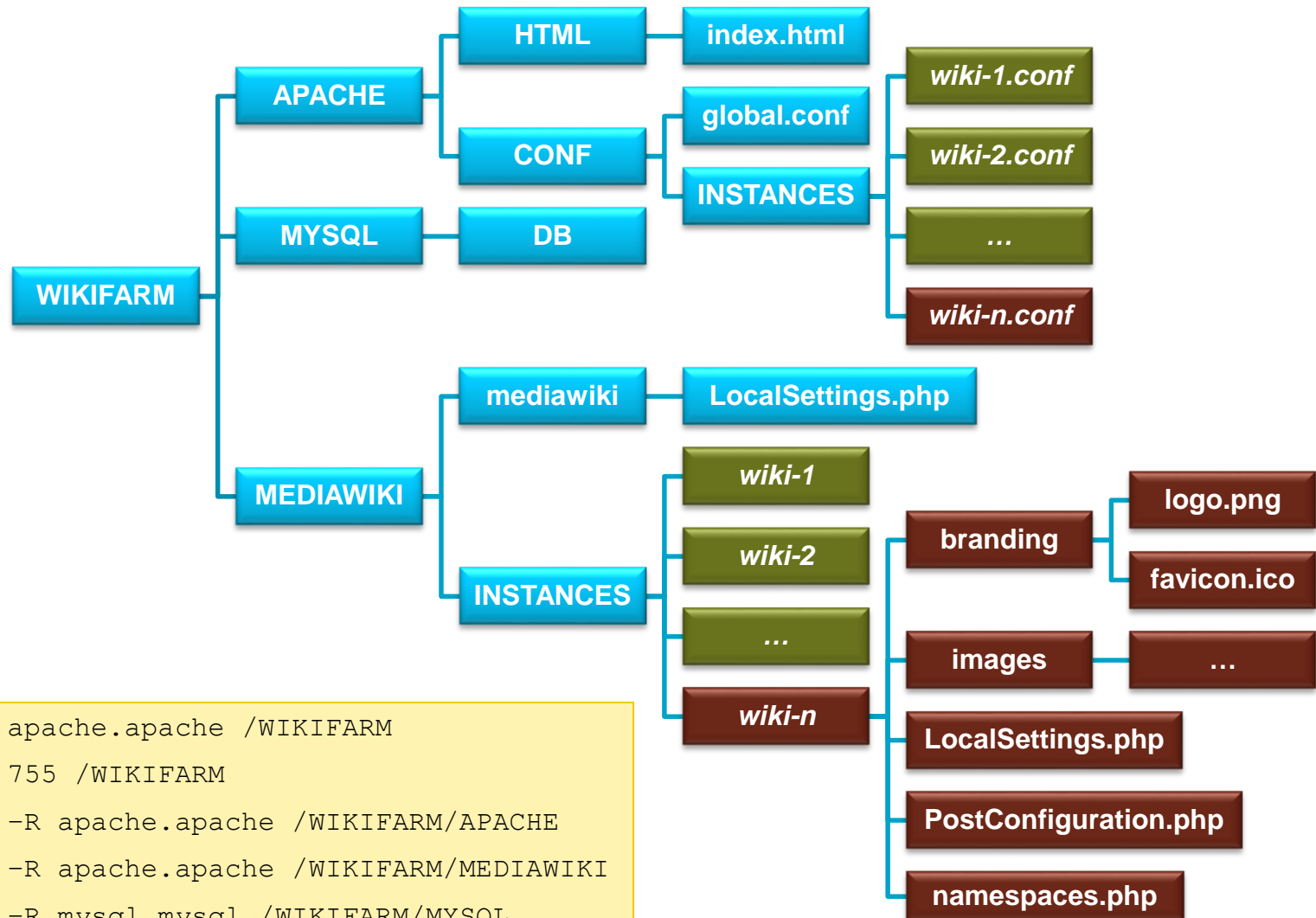
- **Install OS software and dependencies**
- **Set up directory hierarchy**
- **Configure Apache**
- **Configure MySQL**
- **Install MediaWiki**
- **Install extensions**
- **Create a wiki**
  - Create Apache instance directory
  - Create MySQL schema
  - Create MediaWiki instance directory

# Install OS software and dependencies

- Here's what I start with (not necessarily exhaustive):

```
yum -y install polycoreutils-python php php-xml  
php-mbstring php-pecl-apc mysql mysql-server php-  
mysql httpd git ImageMagick squid
```

# Wiki farm directory hierarchy

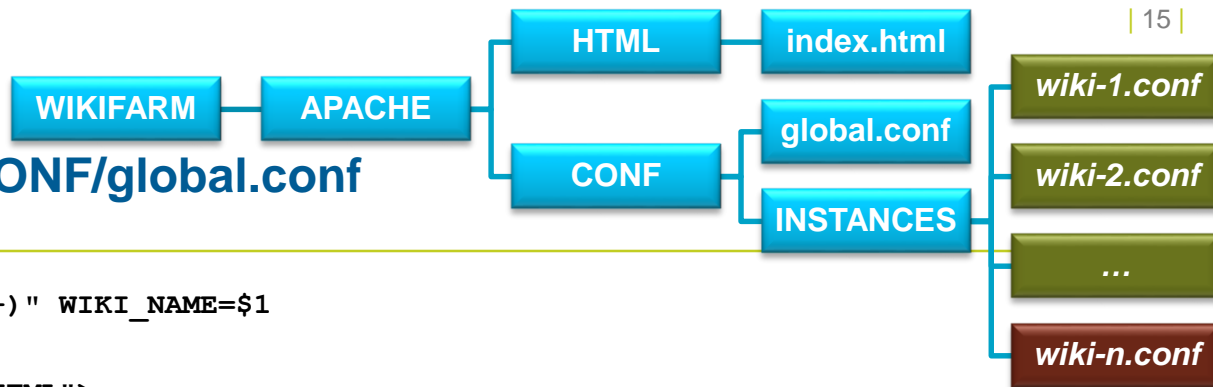


```

chown apache.apache /WIKIFARM
chmod 755 /WIKIFARM
chown -R apache.apache /WIKIFARM/APACHE
chown -R apache.apache /WIKIFARM/MEDIAWIKI
chown -R mysql.mysql /WIKIFARM/MYSQL
  
```

# Create

## /WIKIFARM/APACHE/CONF/global.conf



```
SetEnvIf Request_URI "^/([^/]+)" WIKI_NAME=$1
```

```
<Directory "/WIKIFARM/APACHE/HTML">
    Options +Indexes -FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

```
<Directory "/WIKIFARM/MEDIAWIKI/mediawiki">
    Options +Indexes -FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

```
<Directory "/WIKIFARM/MEDIAWIKI/mediawiki*/mw-config">
    Deny from all
</Directory>
```

```
<Directory "/WIKIFARM/MEDIAWIKI/mediawiki*/images">
    Deny from all
</Directory>
```

```
<Directory "/WIKIFARM/MEDIAWIKI/INSTANCES/*/images">
    Deny from all
```

```
</Directory>
```

© 2015 The MITRE Corporation. All rights reserved.

**WIKI\_NAME**

<http://my.wikifarm.com/mywiki>

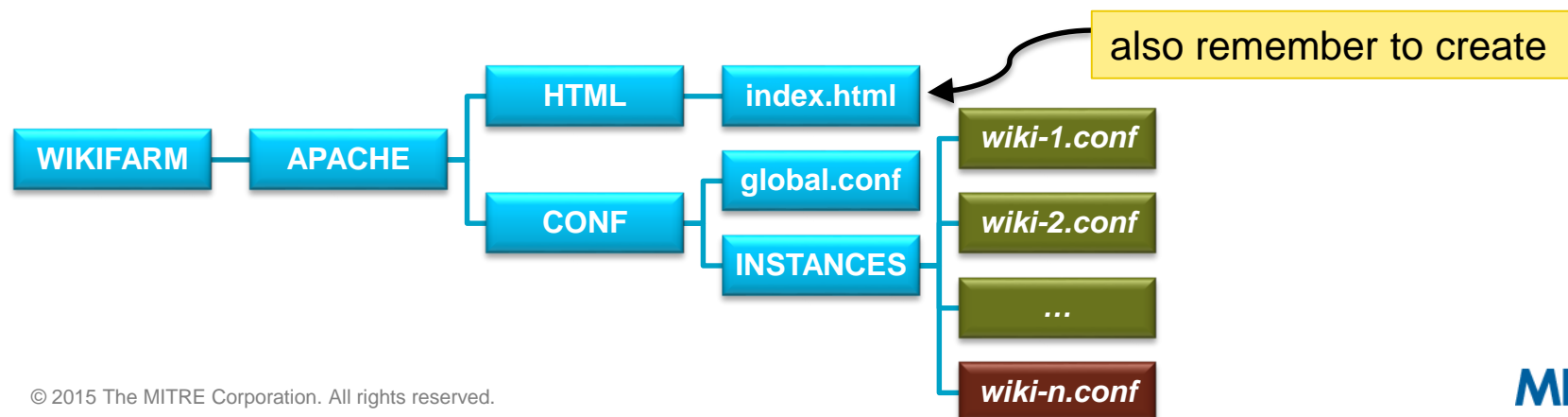
# Edit /etc/httpd/conf/httpd.conf

- Change DocumentRoot from `/var/www/html` to `/WIKIFARM/APACHE/HTML`
- Comment out or remove the Directory blocks referring to `/var/www` and `/var/www/html`

- Add to end:

`Include /WIKIFARM/APACHE/CONF/*.conf`

`Include /WIKIFARM/APACHE/CONF/INSTANCES/*.conf`

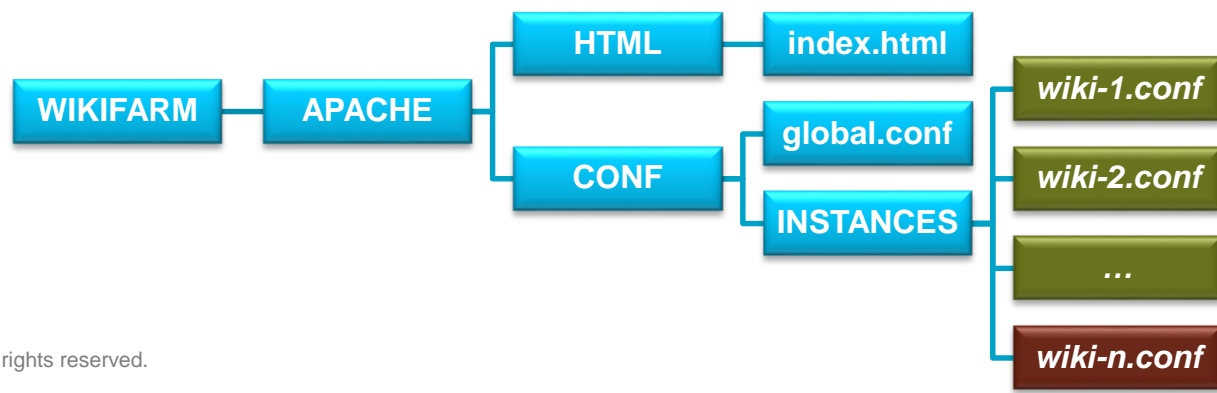




# Create

## /WIKIFARM/APACHE/CONF/INSTANCES/TEMPLATE

```
AliasMatch "^/MyWikiName/branding/(.*)$" /WIKIFARM/MEDIAWIKI/INSTANCES/MyWikiName/branding/$1
AliasMatch "^/MyWikiName$" /WIKIFARM/MEDIAWIKI/mediawiki
AliasMatch "^/MyWikiName/wiki/(.*)$" /WIKIFARM/MEDIAWIKI/mediawiki/index.php/$1
AliasMatch "^/MyWikiName/(.*)$" /WIKIFARM/MEDIAWIKI/mediawiki/$1
<Directory "/WIKIFARM/MEDIAWIKI/INSTANCES/MyWikiName">
    AllowOverride None
    Require all granted
</Directory>
```



# Restart and enable Apache

---

```
service httpd restart  
chkconfig httpd on
```

**OR**

```
systemctl restart httpd.service  
systemctl enable httpd.service
```

# Move MySQL data

WIKIFARM

MYSQL

DB

- **Edit /etc/my.conf:**

- Change:

- `datadir=/var/lib/mysql`

- to

- `datadir=/WIKIFARM/MYSQL/DB`

- Add at the end of the [mysqld] section:

- `bind-address=127.0.0.1`

- **Edit /etc/passwd to change the home directory for the mysql account to /WIKIFARM/MYSQL**

# Restart and enable MySQL

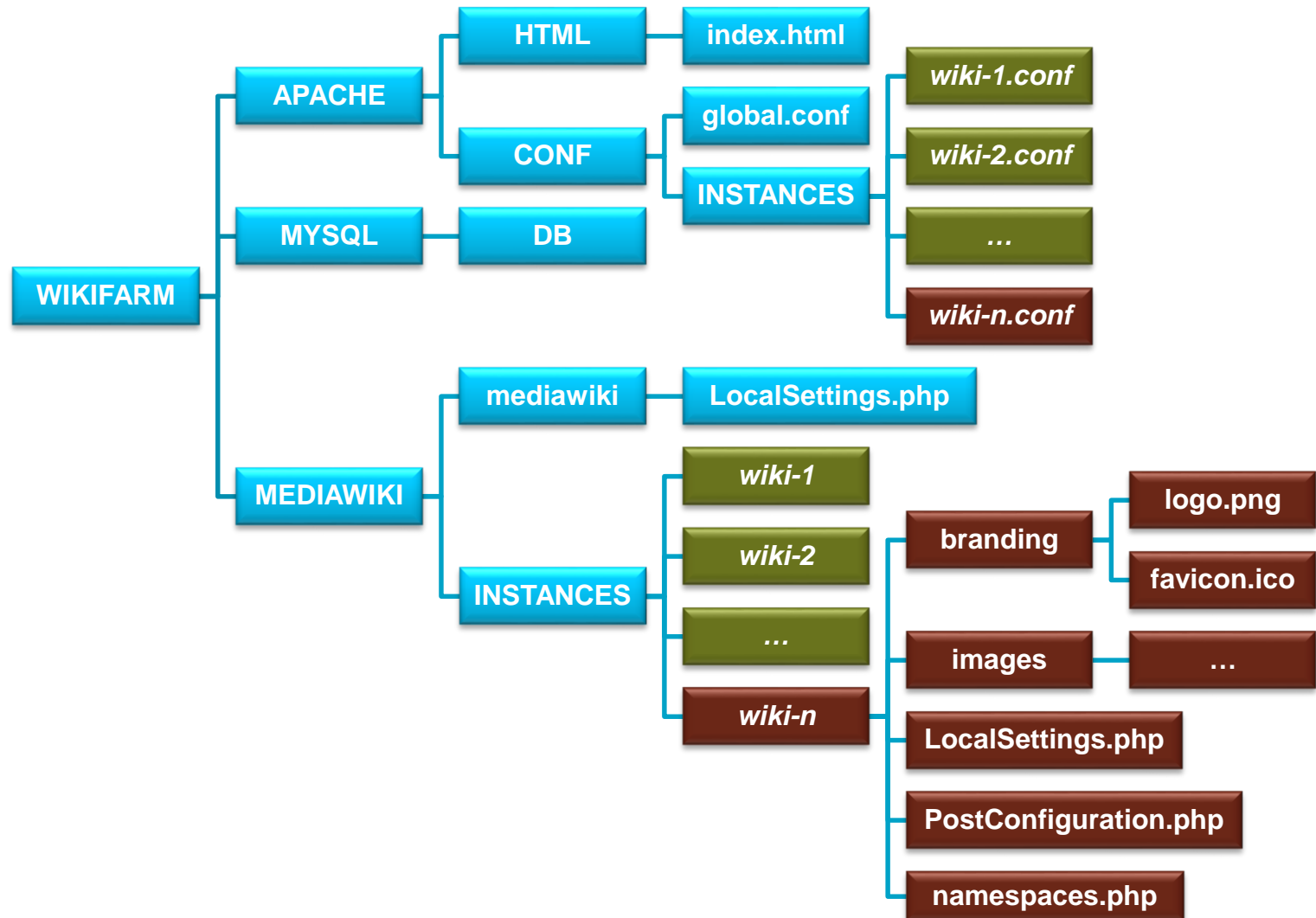
```
service mysqld restart  
chkconfig mysqld on
```

OR

```
systemctl restart mysqld.service  
systemctl enable mysqld.service
```

- **Secure MySQL by running `mysql_secure_installation`**

# Install MediaWiki



# Install MediaWiki

WIKIFARM

MEDIAWIKI

mediawiki

LocalSettings.php

- Clone MediaWiki from git:

```
cd /WIKIFARM/MEDIAWIKI
git clone https://gerrit.wikimedia.org/r/p/mediawiki/core.git mediawiki
```

- List the available release tags:

```
cd mediawiki
git tag -l
```

- Check out the appropriate MediaWiki version:

```
git checkout 1.24.2
```

- Place a copy of LocalSettings.php in /WIKIFARM/MEDIAWIKI/mediawiki
  - saved earlier or created with mw-config

# Edit LocalSettings.php

## - beginning of file:

WIKIFARM

MEDIAWIKI

INSTANCES

wiki-n

branding

logo.png

favicon.ico

images

...

LocalSettings.php

PostConfiguration.php

namespaces.php

```
if (isset($_SERVER['WIKI_NAME'])) {
    $wgSitename = $_SERVER['WIKI_NAME'];
} else {
    die("WIKI_NAME not set.");
}

$wgMetaNamespace = $wgSitename;
$INSTANCE_DIR = "/WIKIFARM/MEDIAWIKI/INSTANCES/$wgSitename";

$wgScriptPath = "/$wgSitename";

$wgServer = "http://my.wikifarm.com";

if( file_exists( "$INSTANCE_DIR/branding/logo.png" ) ) {
    $wgLogo = "/$wgSitename/branding/logo.png";
}

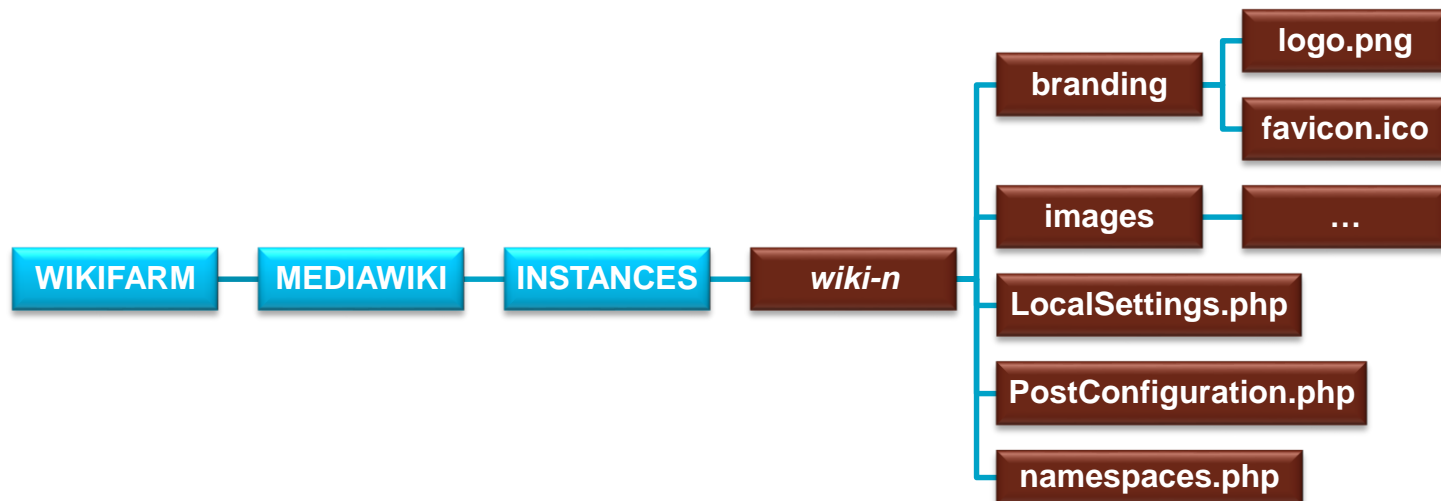
if( file_exists( "$INSTANCE_DIR/branding/favicon.ico" ) ) {
    $wgFavicon = "/$wgSitename/branding/favicon.ico";
}

$wgArticlePath = "/$wgSitename/wiki/$1";
```

# Edit LocalSettings.php

## - enable image authentication

```
$wgUploadPath = "$wgScriptPath/img_auth.php";  
$wgImgAuthPublicTest = false;  
$wgUploadDirectory = "$INSTANCE_DIR/images";  
$wgTmpDirectory = "$INSTANCE_DIR/images/temp";
```





# Edit LocalSettings.php

## - end of file

```
## invoke instance LocalSettings.php
if( file_exists( "$INSTANCE_DIR/LocalSettings.php" ) ) {
    require_once( "$INSTANCE_DIR/LocalSettings.php" );
}

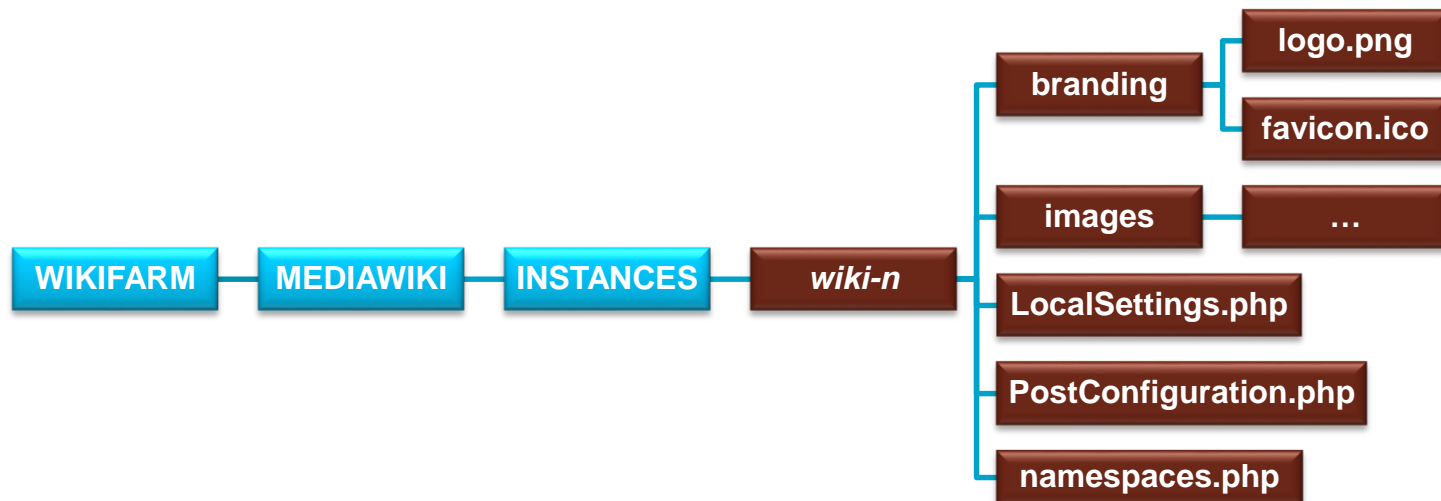
#####
#####
# EXTENSIONS AND EXTENSION CONFIGURATION SETTINGS GO HERE.      #
#####
#####

## invoke instance PostConfiguration.php
if( file_exists( "$INSTANCE_DIR/PostConfiguration.php" ) ) {
    require_once( "$INSTANCE_DIR/PostConfiguration.php" );
}

#####
#####
# DO NOT ADD ANYTHIN ELSE DOWN HERE.                              #
#####
#####
```

# More on instance configuration files

- Allow customization both before and after configuration of global extensions
- Can include additional extensions or override global configuration defaults
- Can define custom namespaces
  - We often put namespace configuration in a separate namespaces.php file included by PostConfiguration.php



# Final MediaWiki installation steps

---

- **Install extensions and add them to LocalSettings.php in the appropriate place**
- **Make sure that all files have their ownership set correctly**
- **Create your wikis**

# Create a wiki

## - create database schema

- First wiki on wiki farm: unblock “mw-config” in Apache and navigate to <http://my.wikifarm.com/mywiki/mw-config>

OR

- In MySQL:
  - if it does not already exist, create a user accessible from localhost for use by MediaWiki and assign it a password:

```
create user [REDACTED]@localhost identified by '[REDACTED]';
```

- create wiki:

```
grant [REDACTED] on mywiki.* to [REDACTED]@localhost;  
create schema mywiki;  
use mywiki;  
source emptywiki.sql; ← saved earlier
```

# Create a wiki

## - configure Apache

---

```
cd /WIKIFARM/APACHE/CONF/INSTANCES
cp TEMPLATE <NEW_WIKI_NAME>.conf
chown apache.apache <NEW_WIKI_NAME>.conf
vim <NEW_WIKI_NAME>.conf ← global replace of MyWikiName
service httpd restart
```

# Create a wiki

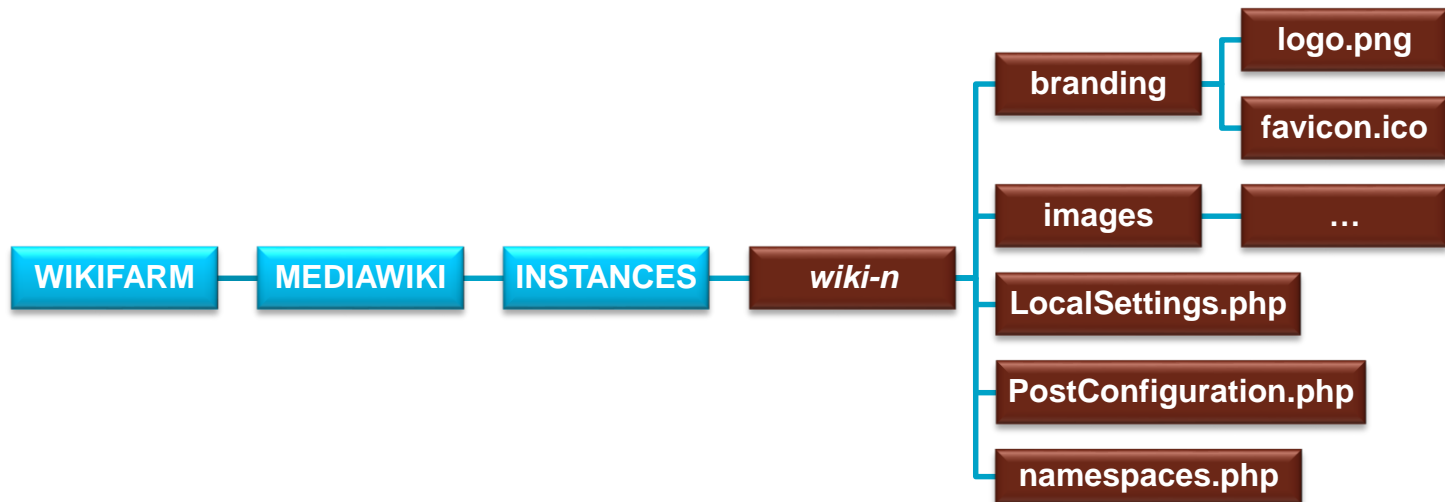
## - create directories

---

```
cd /WIKIFARM/MEDIAWIKI/INSTANCES
mkdir <NEW_WIKI_NAME>
mkdir <NEW_WIKI_NAME>/branding
cp /path/to/logo.png <NEW_WIKI_NAME>/branding
mkdir <NEW_WIKI_NAME>/images
mkdir <NEW_WIKI_NAME>/images/temp
mkdir <NEW_WIKI_NAME>/images/thumb
chown -R apache.apache <NEW_WIKI_NAME>
```

# Create a wiki

## - edit instance configuration files



# Example namespaces.php

```
<?php
```

```
define("NS_MWSTAKE_BLOG", 3000);  
define("NS_MWSTAKE_BLOG_TALK", 3001);  
define("NS_MWSTAKE_EVENT", 3002);  
define("NS_MWSTAKE_EVENT_TALK", 3003);
```

```
$wgExtraNamespaces[NS_MWSTAKE_BLOG] = "Blog_Post";  
$wgExtraNamespaces[NS_MWSTAKE_BLOG_TALK] = "Blog_Post_talk";  
$wgExtraNamespaces[NS_MWSTAKE_EVENT] = "Event";  
$wgExtraNamespaces[NS_MWSTAKE_EVENT_TALK] = "Event_talk";
```

```
$wgContentNamespaces[] = NS_MWSTAKE_BLOG;  
$wgContentNamespaces[] = NS_MWSTAKE_EVENT;
```

```
$wgNamespacesToBeSearchedDefault[NS_MWSTAKE_BLOG] = true;  
$wgNamespacesToBeSearchedDefault[NS_MWSTAKE_EVENT] = true;
```

```
$smwgNamespacesWithSemanticLinks[NS_MWSTAKE_BLOG] = true;  
$smwgNamespacesWithSemanticLinks[NS_MWSTAKE_EVENT] = true;
```



# Example PostConfiguration.php

```
<?php

$wgCustomNavBlocksEnable = true;

require_once("namespaces.php");

$wgSemanticTitleHideSubtitle = true;
$wgSemanticTitleProperties = array(
    NS_MWSTAKE_BLOG => 'Name',
    NS_MWSTAKE_EVENT => 'Name',
    NS_USER => 'Name'
);
```

# Create a wiki

## - run scripts if necessary

---

```
cd /WIKIFARM/MEDIAWIKI/mediawiki
export WIKI_NAME=<WIKI_NAME>
php maintenance/update.php
php maintenance/refreshLinks.php
php maintenance/runJobs.php
cd /WIKIFARM/MEDIAWIKI/INSTANCES/<WIKI_NAME>
chown -R apache.apache images
```

# Additional configuration (details not discussed today)

---

- **Squid**
- **SELinux**
- **SSL**
- **Additional security configuration**
- **Install web services (Parsoid, Mathoid, etc.)**
- **OS configuration and additional tool support**
- **With minor configuration changes, this setup can also be tweaked to support single wiki servers without requiring the additional wiki name piece in the URL**

# Future directions

---

- **Scripted wiki farm creation**
  - Ansible
- **Scripted wiki creation**
- **Drag and drop semantic modeling tool**